RECOMMENDATION SYSTEM USING MACHINE LEARNING ALGORITHMS

Gaurav Singh[1], Mayank Upadhyay[2], Umang Sharma[3], Shahrukh Hussain[4], Amit Pandey[5]

[1,2,3,4]Student, [5]Assistant Professor, Department of Information Technology,
Dr. Akhilesh Das Gupta Institute of Technology and Management

---***---

Abstract - *Today Artificial Intelligence has been a blooming Technology which aims at doing things that traditionally require human intelligence. In the field of Artificial Intelligence, Machine Learning has proved its mettle in building such automated system that is once trained and then the machine learns from experience just like humans which does not require any explicit programming. Machine Learning & Artificial intelligence is now a very important because they have huge potential for all its applications across various industries be it Healthcare, Ed Tech, Manufacturing and many more. Machine Learning are now contributing to like every industry and whichever company has shown a rapid growth, it has not been possible without the use of Artificial Intelligence and Machine Learning Algorithms. One such important area in field of Artificial Intelligence and Machine Learning is building Recommendation System. In this the research work has been done in building a movie recommender system using Machine Learning Algorithm namely K-Nearest Neighbor. The dataset which have been used in building recommendation system is movie lens dataset from Kaggle. The proposed work is aligned with the various concepts associated with machine learning algorithms and recommendation system. In this work we have used various libraries like pandas, algorithms like KNN, techniques like Collaborative Filtering, Content-Based Filtering which have been discussed in detail. Thus after getting a clear view of various techniques, we found which algorithm to be applied in different areas of industries. Finally, an algorithm is devised that can be implemented for building a recommended system.*

Key Words: Recommendation System, Machine Learning, KNN, Collaborative Filtering, Content-Based Filtering,

1. INTRODUCTION

In Today's hectic world Recommendation system has become very important part of any service that require user interaction between other users and item. This is very much needed as people always want product and services that suits them best. Thus Recommendation system are now very important as they help in making the right choices and get a reach of all the similar products. [1]

In this research work, we will deep dive into all concepts related to Recommendation Systems and learn how can a Movie Recommendation System can be built using collaborative filtering by using the K-Nearest Neighbors algorithm.

Types of Recommendation Systems

Recommendation systems can be mainly classified into 3 different types: [1]

1. Collaborative Filtering
2. Content-Based Filtering
3. Hybrid Recommendation Systems

Collaborative Filtering

Collaborative filtering generally based on gathering and analyzing the data and information on user's behaviors and choices, their preferences or their activities and then making predictions what will they like based on similarity with the other users of nearly same choice and preference. [2] The major advantage of collaborative filtering approach is that it is not depend totally on machine learning and analyzable content and so it makes it capable of making accurate recommendations even if involving highly complicated items without analyzing each item itself. [10]

Several types of collaborative filtering algorithms are described here:

- **User-User Collaborative Filtering:** In this kind of filtering, we try to search the customers that are almost lookalike that is the customers that are most similar to each other will be grouped together. [3]
- **Item-Item Collaborative Filtering:** In this kind of filtering, we try to search the item that are almost similar or items that are together used, if the customer goes for an item then all other items that are used along with that item is recommended to the customer. [3]

Content based Filtering

Usually this kind of filtering approaches are based on the item description and likes, choices and preference of user. Thus in content based recommendation, there are keywords related to an item and based on that if user prefer this items than these keywords get associated with that user and thus build a user profile. [8] These items are recommended to user. In easy words, this algorithm will recommend the items to the user that are most similar with the items that the user has liked in past. [10]

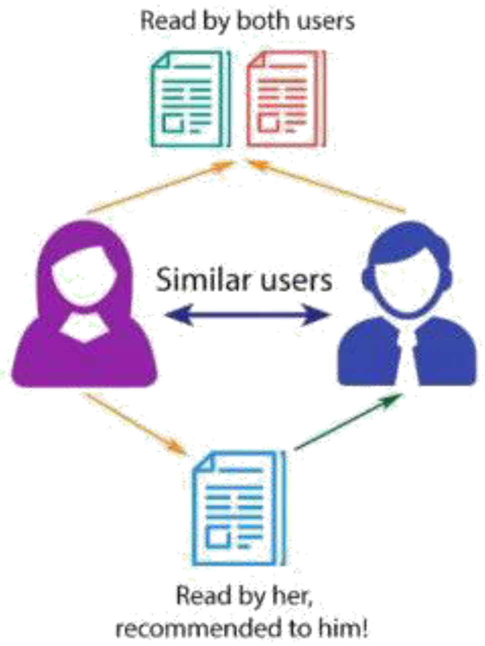


Fig -1: Collaborative filtering Recommendation System illustration

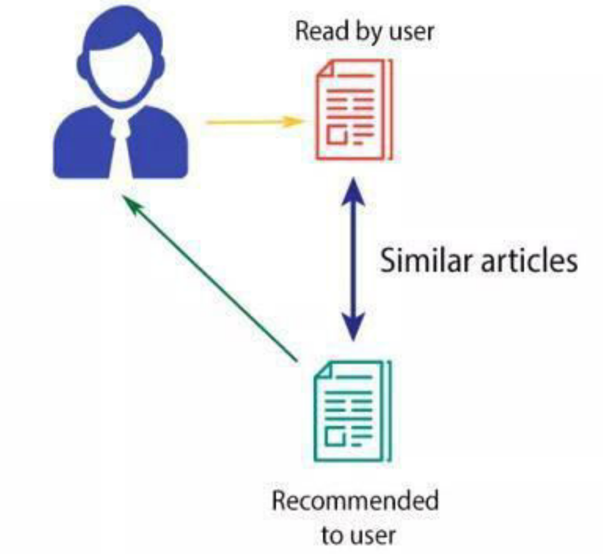


Fig -2: Content-Based filtering Recommendation System illustration [8]

Hybrid Recommendation Systems

There have been many live examples today that a hybrid approach, which is done by combining collaborative filtering and a content based filtering can give more accurate and satisfying results. [3] There are several ways to implement hybrid algorithms, by adding capabilities of content based algorithms to the collaborative based filtering algorithms or vice versa, by

separately running content based algorithm and collaborative based filtering algorithm and then combine the result, or it can be by combine both the algorithm and make one model out of it. [3] [10]

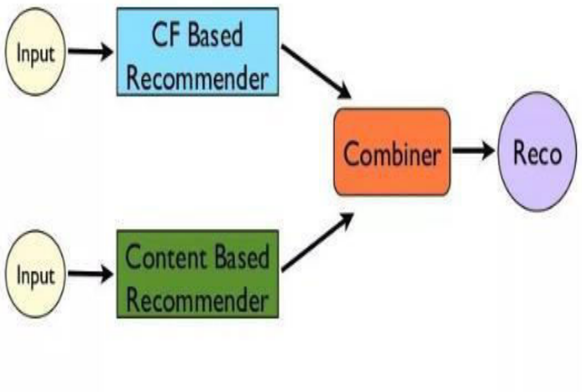


Fig -3: Hybrid Recommendation System illustration [8]

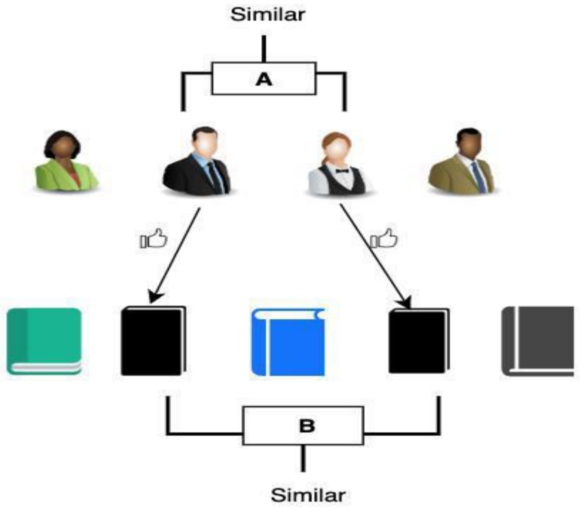


Fig -4: How Hybrid System works [8]

2. LITERATURE REVIEW

All the three recommendation methods, be it collaborative based filtering algorithm content based algorithm, hybrid recommendation, have been used by many past researchers to build and develop a more reliable recommendation system. A movie recommendation system which was a web based was built using hybrid filtering methods by the authors in 2007. [4] [9] Another movie recommendation system is proposed by the authors in 2011, [4] [9] which was based on genre correlations. Also in 2013 there was one more recommendation system which was proposed and was based on Bayesian network. Then in 2016 there was another recommendation system proposed that uses rating and scores for the users and items that lead to one more kind of recommendation system. In today's world Recommendation system has become a integral part of all social networking sites like Amazon, Facebook, Netflix, YouTube, Twitter and many more and with effect of which both the customers, [10]users, and the company itself has raised their progress exponentially.

3. PROPOSED WORK

3.1 Data

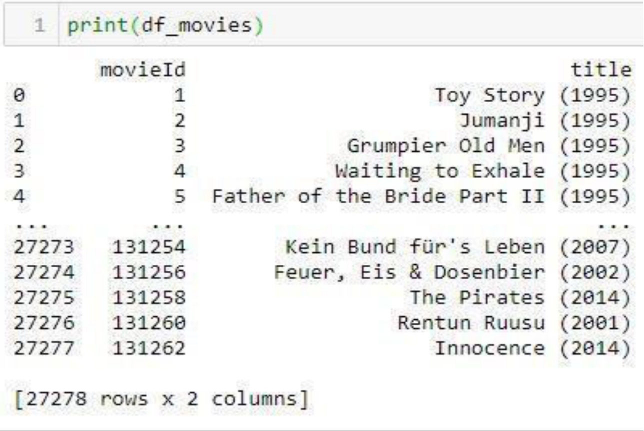
The Dataset which we have used for this work is Movielens Dataset. This is a very popular and widely sued Dataset for building recommendation system. This dataset comes with various invariants which are of different sizes that is 100k dataset, 1M dataset, 2M dataset, 20M dataset. For your work we have used 20M dataset. In this data. There are two main csv files as movies.csv which include the title, genre of the movie and rating.csv which include movieID and the rating associated with it. [5]

3.1.1 Importing the libraries/ Dataset

Importing the essential libraries such as Numpy, Pandas, Matplotlib, Seaborn. The movielens dataset is loaded and read using pandas library.

3.1.2 Data carpentary/ Cleaning

There are several columns that can have NULL values or some of rows have some fields as null values. So, we have to process the data, for that we can use inbuilt methods for this.



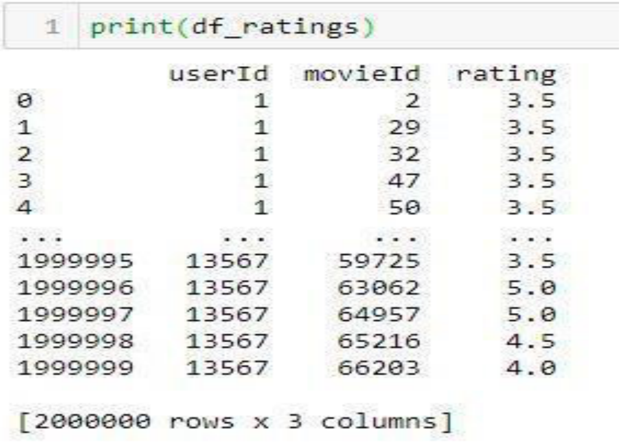
```
print(df_movies)
```

```
       movieId                               title
0            1                    Toy Story (1995)
1            2                      Jumanji (1995)
2            3             Grumpier Old Men (1995)
3            4            Waiting to Exhale (1995)
4            5  Father of the Bride Part II (1995)
...        ...                                 ...
27273   131254         Kein Bund für's Leben (2007)
27274   131256        Feuer, Eis & Dosenbier (2002)
27275   131258                  The Pirates (2014)
27276   131260                 Rentun Ruusu (2001)
27277   131262                   Innocence (2014)

[27278 rows x 2 columns]
```

Fig -5: Movies.csv dataset

The values in rating.csv can be NULL, so we will fill 0s wherever the data value is NULL.



```
print(df_ratings)
```

```
         userId  movieId  rating
0             1        2     3.5
1             1       29     3.5
2             1       32     3.5
3             1       47     3.5
4             1       50     3.5
...         ...      ...     ...
1999995   13567    59725     3.5
1999996   13567    63062     5.0
1999997   13567    64957     5.0
1999998   13567    65216     4.5
1999999   13567    66203     4.0

[2000000 rows x 3 columns]
```

Fig -6: Ratings.csv dataset

3.1.3 Identifying Threshold values

It can be seen there are 0s values in our model data. Thus we are dealing with a sparse data (Fig-6). Secondly, the algorithm can only work well for a user if he has rated atleast 50 movies and moreover the algorithm also work well when any movie has been rate for atleast 50 times. [5]

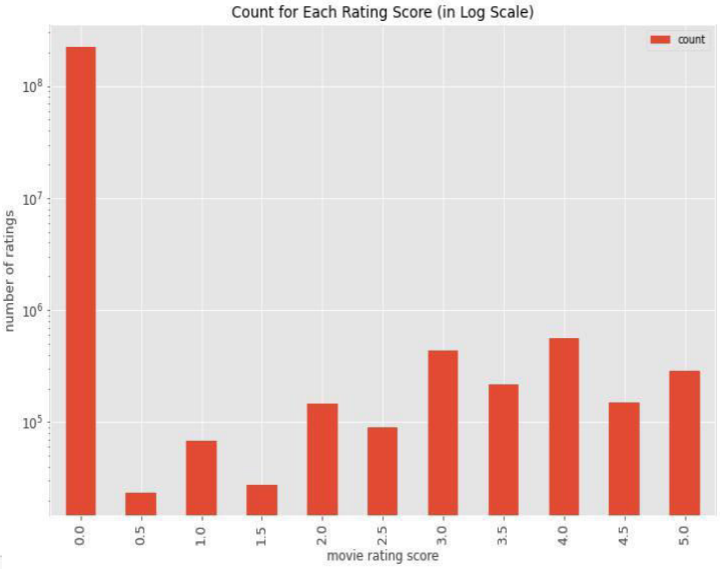


Fig -7: Rating: log(count) [5]

3.2 Algorithm

In this work we have taken the use of K-Nearest Neighbor Algorithm. In KNN algorithm, for a particular data point, the K nearest point with a specific distance which is calculated by a specific distance formula. Now if the value of K becomes 1, then the first nearest point to that point is the same point itself. [7]

3.2.1 Distance Formulas:

1) For Continuous Values: There are three distance metrics for continuous values namely, Euclidean, Manhattan, Minkowski.[5] [7]

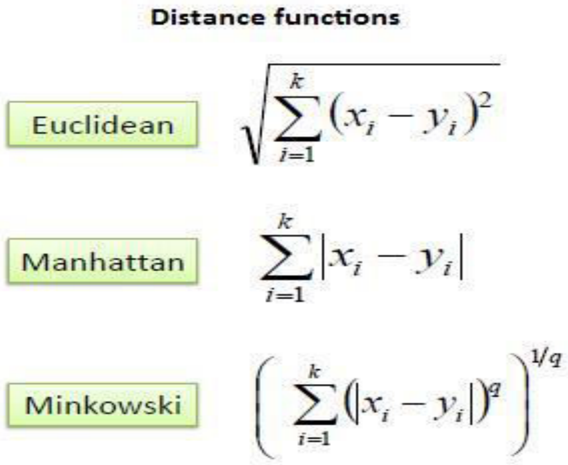

Distance functions
Euclidean
$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$
Manhattan
$\sum_{i=1}^{k}|x_i - y_i|$
Minkowski
$\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

Fig -8: Distance Functions[5]

2) For Categorical Values: For categorical data we have one and very famous way of finding distance that

is Hamming distance.[5][7]

Hamming Distance

$$D_H = \sum_{i=1}^{k} |x_i - y_i|$$

$$x = y \Rightarrow D = 0$$

$$x \neq y \Rightarrow D = 1$$

Fig -8: Distance Functions [5]

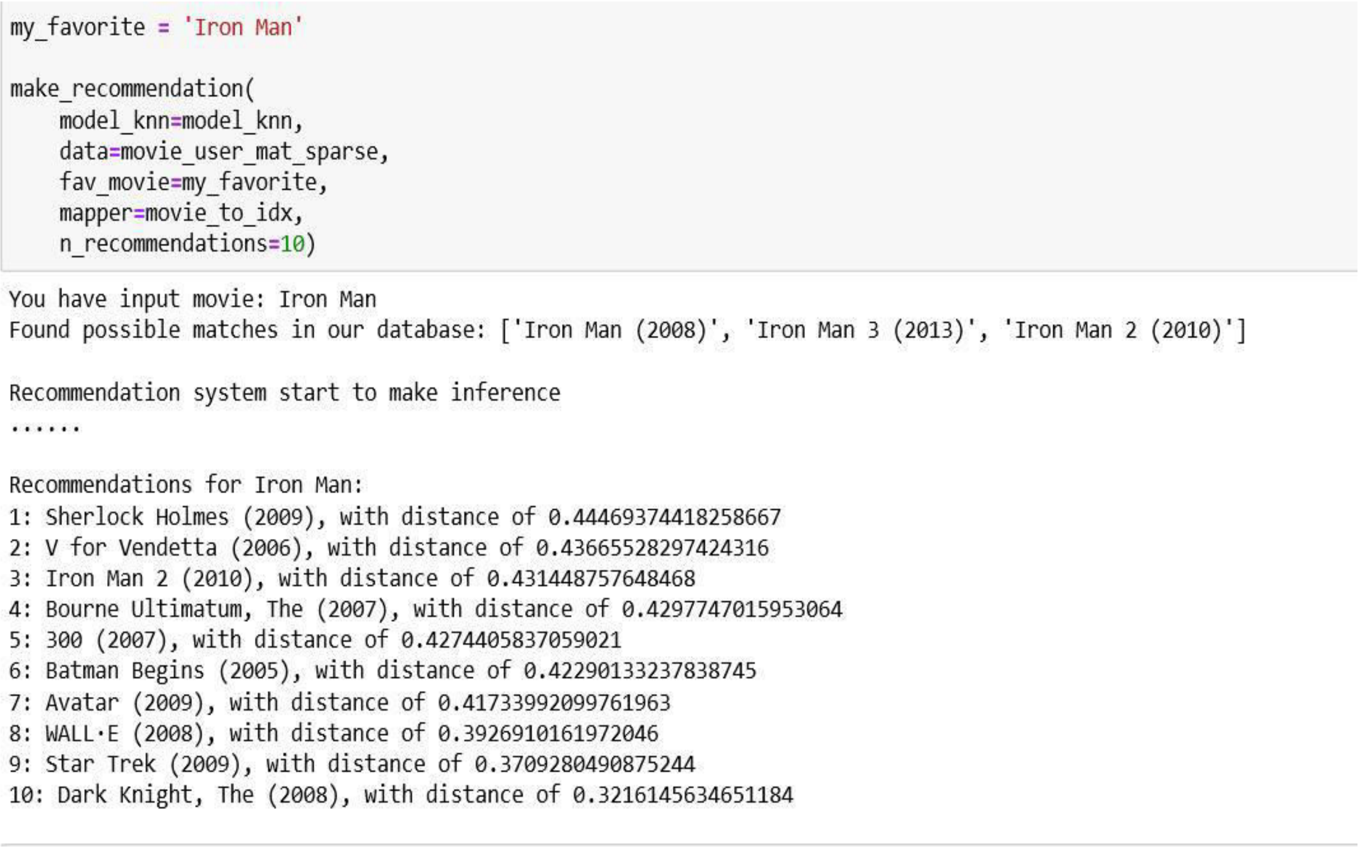
3.3 Running Algorithm/ Making Recommendation

- Choose any unclassified point in the n-D space.
- Compute the distance using any if the distance formula depending on the value associativity. Calculate the required distance from the chosen unclassified point to all other points available in the space.
- Apply KNN to find the nearest K points [5] corresponding to the chosen unclassified point.
- Find the frequency of each class for these points.
- Whichever class turn out to be with the largest count would be the answer for that chosen point and that would be used by the algorithm.
- After finding the class for that data point , assign that particular class to the chosen point.
- Thus repeat these steps for all the data points, when the algorithm training is over.

The proposed algorithm (Fig-9) is able to recommend the movie for a arbitrary movie. We will take the movie name from the user, then find the movie Id for that particular movie, then we will give that movie Id to our model, our proposed algorithm will recommend us to top K movies which belong to the same genre/ category of the movie we have chosen. [5]

4 RESULT AND DISCUSSION

Here (fig-9) it can be seen that the if the movie name or the movie ID is given as the input, then the top N movies are recommended which are most similar with genre and category which the viewers watch. Here we have given the input movie name as "Iron Man", which is fictional, superhero, action and thrill movie. Then the top 10 movies which are recommended are of this kind only. But the important thing here is we have chosen a movie that has already been rated by so many viewers, so we were able to recommend movie correctly. Also if we want to recommend the movie for a particular user, then also the restriction is that the user has recommended a good number of movies. Thus recommendation system are efficient only when the interaction of movies and the users is high, That means if a new movie or a new user is added to system, initially they both won't create much difference to previous model. So this is how we proposed a movie recommender system.



```
my_favorite = 'Iron Man'

make_recommendation(
    model_knn=model_knn,
    data=movie_user_mat_sparse,
    fav_movie=my_favorite,
    mapper=movie_to_idx,
    n_recommendations=10)
```

```
You have input movie: Iron Man
Found possible matches in our database: ['Iron Man (2008)', 'Iron Man 3 (2013)', 'Iron Man 2 (2010)']

Recommendation system start to make inference
......

Recommendations for Iron Man:
1: Sherlock Holmes (2009), with distance of 0.44469374418258667
2: V for Vendetta (2006), with distance of 0.43665528297424316
3: Iron Man 2 (2010), with distance of 0.431448757648468
4: Bourne Ultimatum, The (2007), with distance of 0.4297747015953064
5: 300 (2007), with distance of 0.4274405837059021
6: Batman Begins (2005), with distance of 0.42290133237838745
7: Avatar (2009), with distance of 0.41733992099761963
8: WALL·E (2008), with distance of 0.3926910161972046
9: Star Trek (2009), with distance of 0.3709280490875244
10: Dark Knight, The (2008), with distance of 0.3216145634651184
```

Fig -9: Result of making the recommendation.

5.CONCLUSION

Machine Learning has proven itself in analyzing data which can help in automating the programing systems that require manual intervention earlier. The machine learning algorithms learns from data and develop some mathematical graphs or clusters such that if new data point is given to ML model, it can clearly compute/predict that what is the category or the value of this new data point. Similar is done in our proposed work also, we have used KNN to identify the most similar items/ movies corresponding to the given movie and thus successfully recommended the user all the movies of his choices. The algorithm in the proposed work can also be improved using more and more data sets and finding out the proximal values. In future, various characteristic for a particular item can be used which are latent in the recommendation items to users or within between users.

REFERENCES

[1] HTTPS://WWW.RESEARCHGATE.NET/PUBLICATION/334763301_MOVIE_RECOMMENDER_SYSTEM_USING_K-MEANS_CLUSTERING_AND_K-NEAREST_NEIGHBOR

[2] M VISWA MURALI; T G VISHNU; NANCY VICTOR, A COLLABORATIVE FILTERING BASED RECOMMENDER SYSTEM FOR SUGGESTING NEW TRENDS IN ANY DOMAIN OF RESEARCH, IEEE, 2020 MAR 15

[3] HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/2020/08/RECOMMENDATION-SYSTEM-K-NEAREST-NEIGHBORS/

[4] SHREYA AGRAWAL, POOJA JAIN, AN IMPROVED APPROACH FOR MOVIE RECOMMENDATION SYSTEM, IEEE, 2020 FEB 10

[5] HTTPS://HEARTBEAT.FRITZ.AI/RECOMMENDER-SYSTEMS-WITH-PYTHON-PART-II-COLLABORATIVE-FILTERING-K-NEAREST-NEIGHBORS-ALGORITHM-C8DCD5FD89B2

[6] MEENU GUPTA; ADITYA THAKKAR; AASHISH; VISHAL GUPTA; DHRUV PRATAP SINGH RATHORE, MOVIE RECOMMENDER SYSTEM USING COLLABORATIVE FILTERING, IEEE 2020 JUL 4

[7] HTTPS://TOWARDSDATASCIENCE.COM/PROTOTYPING-A-RECOMMENDER-SYSTEM-STEP-BY-STEP-PART-1-KNN-ITEM-BASED-COLLABORATIVE-FILTERING-637969614EA

[8] HTTPS://TOWARDSDATASCIENCE.COM/THE-REMARKABLE-WORLD-OF-RECOMMENDER-SYSTEMS-BFF4B9CBE6A7

[9] HTTPS://REALPYTHON.COM/BUILD-RECOMMENDATION-ENGINE-COLLABORATIVE-FILTERING/

[10] HTTP://CS229.STANFORD.EDU/PROJ2019AUT/DATA/ASSIGNMENT_308832_RAW/26588542.PDF

[11] HTTPS://TOWARDSDATASCIENCE.COM/PROTOTYPING-A-RECOMMENDER-SYSTEM-STEP-BY-STEP-PART-1-KNN-ITEM-BASED-COLLABORATIVE-FILTERING-637969614EA